Sql - Basics

1) **Creating Database:**

   The CREATE DATABASE statement is used to create a new database to store your tables and data.

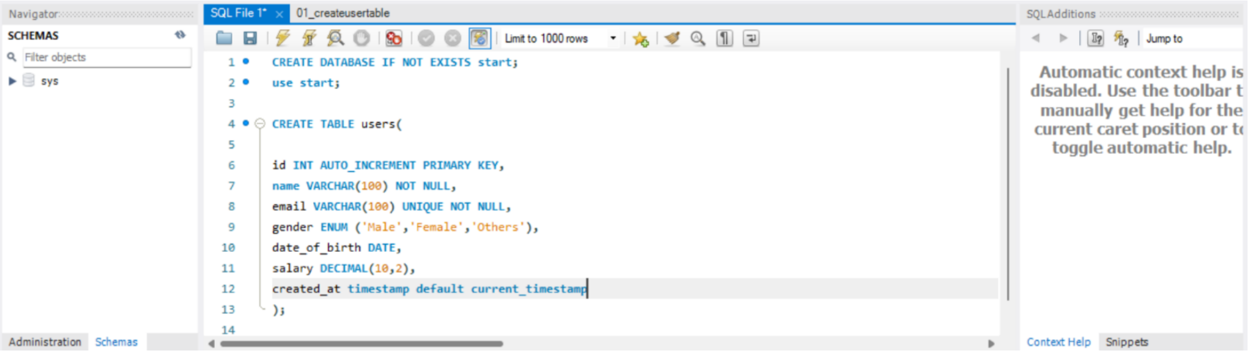
   Query: CREATE DATABASE IF NOT EXISTS database_name;

2) **Creating Tables:**

   The CREATE TABLE statement is used to create a new table in the database by defining column names and their data types.

   Query:

```
  CREATE TABEL table_name (
    Column_name datatype constraint (id INT AUTO_INCREMENT PRIMARY KEY)
)
```



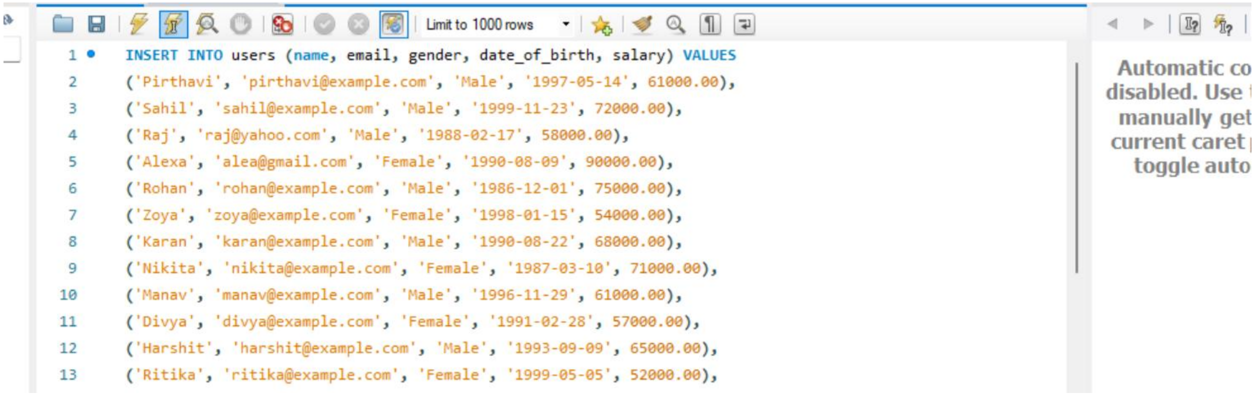
```
CREATE DATABASE IF NOT EXISTS start;
use start;

CREATE TABLE users(

id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
gender ENUM ('Male','Female','Others'),
date_of_birth DATE,
salary DECIMAL(10,2),
created_at timestamp default current_timestamp
);
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

3) Inserting Data into Tables

The INSERT INTO statement adds new records (rows) into a table.
We can insert data into all columns or specify particular ones.

Query: INSERT INTO users (name, email, gender, date_of_birth, salary) VALUES
('Pirthavi', 'pirthavi@gmail.com', 'Male', '1997-08-01', 60000);



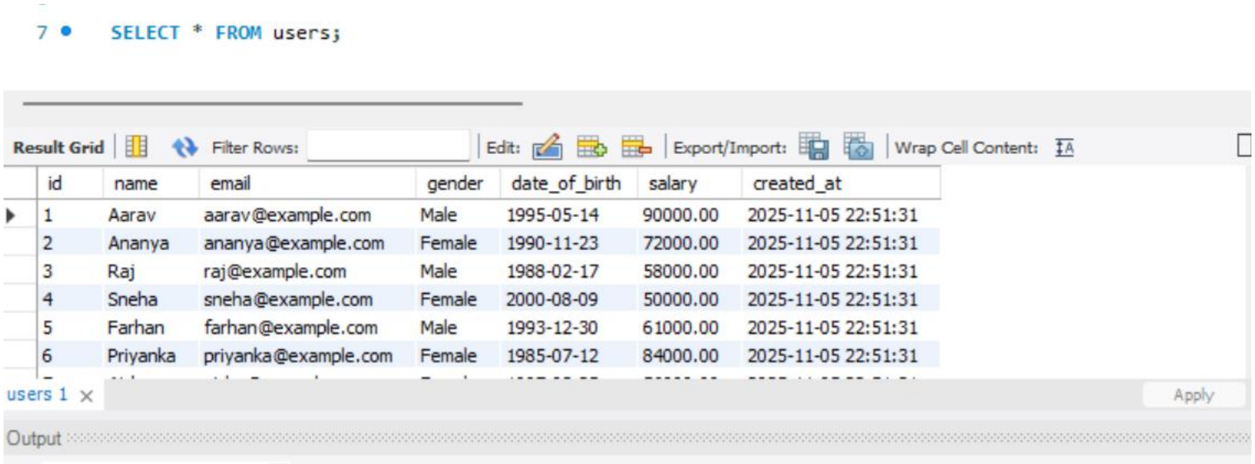
```
INSERT INTO users (name, email, gender, date_of_birth, salary) VALUES
('Pirthavi', 'pirthavi@example.com', 'Male', '1997-05-14', 61000.00),
('Sahil', 'sahil@example.com', 'Male', '1999-11-23', 72000.00),
('Raj', 'raj@yahoo.com', 'Male', '1988-02-17', 58000.00),
('Alexa', 'alea@gmail.com', 'Female', '1990-08-09', 90000.00),
('Rohan', 'rohan@example.com', 'Male', '1986-12-01', 75000.00),
('Zoya', 'zoya@example.com', 'Female', '1998-01-15', 54000.00),
('Karan', 'karan@example.com', 'Male', '1990-08-22', 68000.00),
('Nikita', 'nikita@example.com', 'Female', '1987-03-10', 71000.00),
('Manav', 'manav@example.com', 'Male', '1996-11-29', 61000.00),
('Divya', 'divya@example.com', 'Female', '1991-02-28', 57000.00),
('Harshit', 'harshit@example.com', 'Male', '1993-09-09', 65000.00),
('Ritika', 'ritika@example.com', 'Female', '1999-05-05', 52000.00),
```

4) SELECT

The SELECT statement is used to fetch data from one or more tables in a database. It defines what columns you want to see in the result.

Query: SELECT * FROM users;

Query: SELECT email, gender FROM users;



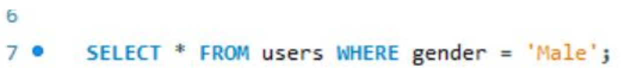
```
SELECT * FROM users;
```

id	name	email	gender	date_of_birth	salary	created_at
1	Aarav	aarav@example.com	Male	1995-05-14	90000.00	2025-11-05 22:51:31
2	Ananya	ananya@example.com	Female	1990-11-23	72000.00	2025-11-05 22:51:31
3	Raj	raj@example.com	Male	1988-02-17	58000.00	2025-11-05 22:51:31
4	Sneha	sneha@example.com	Female	2000-08-09	50000.00	2025-11-05 22:51:31
5	Farhan	farhan@example.com	Male	1993-12-30	61000.00	2025-11-05 22:51:31
6	Priyanka	priyanka@example.com	Female	1985-07-12	84000.00	2025-11-05 22:51:31

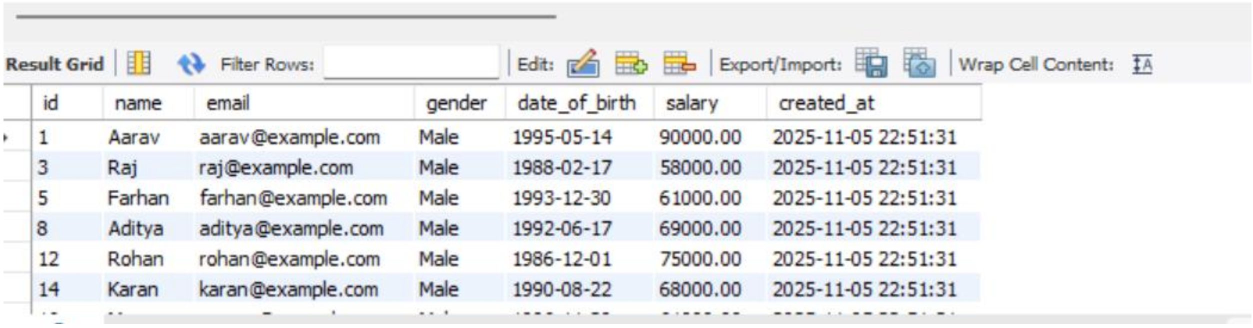
5) WHERE:

The WHERE clause is used to filter records that meet specific conditions. It helps you retrieve only the rows that satisfy a given condition.

Query: SELECT * FROM users WHERE gender = ‘Male’;



```sql
SELECT * FROM users WHERE gender = 'Male';
```



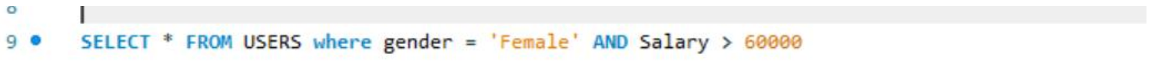
id	name	email	gender	date_of_birth	salary	created_at
1	Aarav	aarav@example.com	Male	1995-05-14	90000.00	2025-11-05 22:51:31
3	Raj	raj@example.com	Male	1988-02-17	58000.00	2025-11-05 22:51:31
5	Farhan	farhan@example.com	Male	1993-12-30	61000.00	2025-11-05 22:51:31
8	Aditya	aditya@example.com	Male	1992-06-17	69000.00	2025-11-05 22:51:31
12	Rohan	rohan@example.com	Male	1986-12-01	75000.00	2025-11-05 22:51:31
14	Karan	karan@example.com	Male	1990-08-22	68000.00	2025-11-05 22:51:31

6) AND / OR

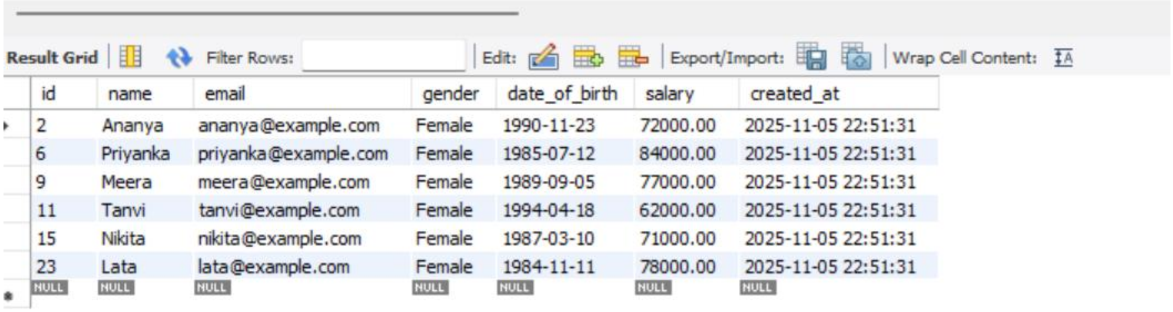
- The AND and OR operators combine multiple conditions in a WHERE clause.
- AND returns records only if all conditions are true.
- OR returns records if at least one condition is true.

Query: SELECT * FROM users WHERE gender = ‘Male’ **AND** salary <60000;

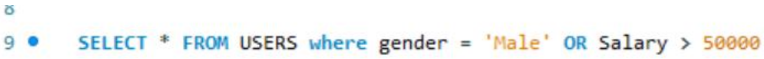
Query: SELECT * FROM users WHERE gender = ‘Male’ **OR** salary <60000;



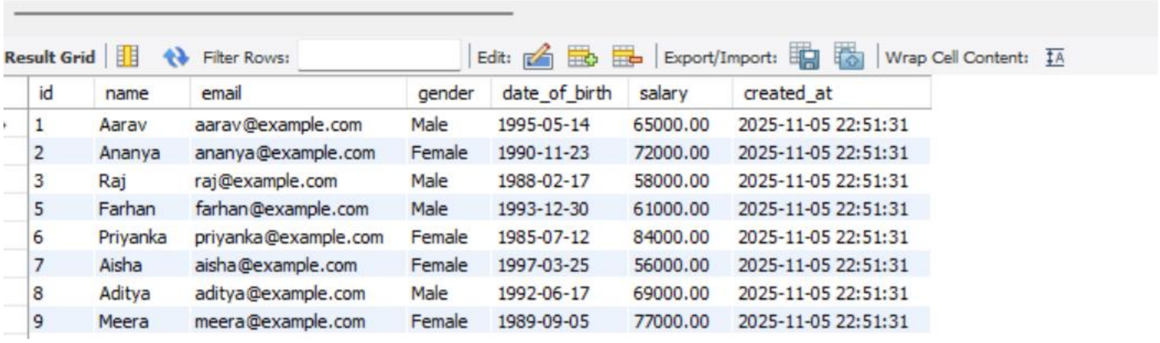
```sql
SELECT * FROM USERS where gender = 'Female' AND Salary > 60000
```



id	name	email	gender	date_of_birth	salary	created_at
2	Ananya	ananya@example.com	Female	1990-11-23	72000.00	2025-11-05 22:51:31
6	Priyanka	priyanka@example.com	Female	1985-07-12	84000.00	2025-11-05 22:51:31
9	Meera	meera@example.com	Female	1989-09-05	77000.00	2025-11-05 22:51:31
11	Tanvi	tanvi@example.com	Female	1994-04-18	62000.00	2025-11-05 22:51:31
15	Nikita	nikita@example.com	Female	1987-03-10	71000.00	2025-11-05 22:51:31
23	Lata	lata@example.com	Female	1984-11-11	78000.00	2025-11-05 22:51:31
NULL	NULL	NULL	NULL	NULL	NULL	NULL



```sql
SELECT * FROM USERS where gender = 'Male' OR Salary > 50000
```

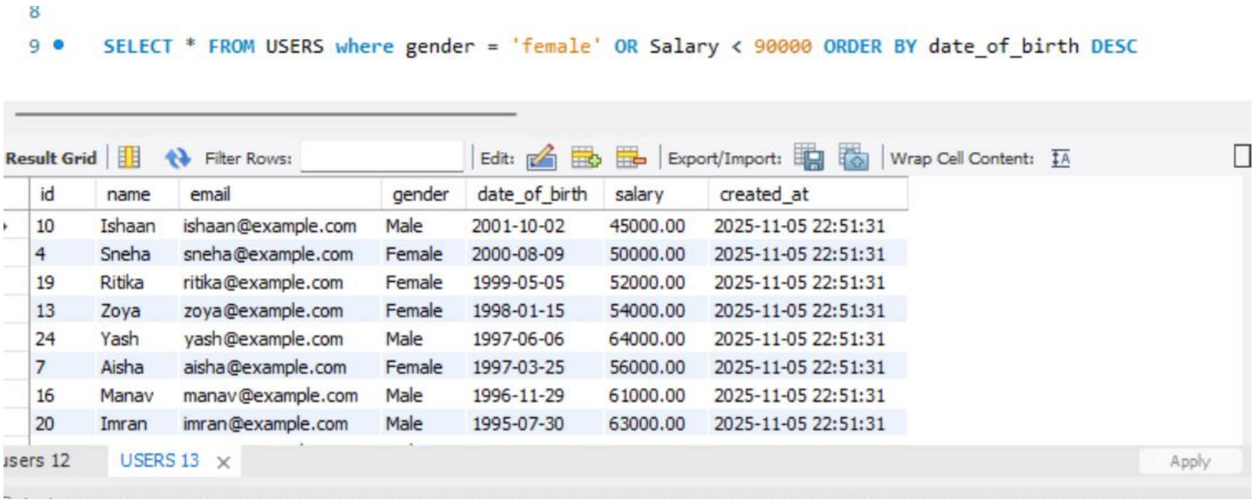


id	name	email	gender	date_of_birth	salary	created_at
1	Aarav	aarav@example.com	Male	1995-05-14	65000.00	2025-11-05 22:51:31
2	Ananya	ananya@example.com	Female	1990-11-23	72000.00	2025-11-05 22:51:31
3	Raj	raj@example.com	Male	1988-02-17	58000.00	2025-11-05 22:51:31
5	Farhan	farhan@example.com	Male	1993-12-30	61000.00	2025-11-05 22:51:31
6	Priyanka	priyanka@example.com	Female	1985-07-12	84000.00	2025-11-05 22:51:31
7	Aisha	aisha@example.com	Female	1997-03-25	56000.00	2025-11-05 22:51:31
8	Aditya	aditya@example.com	Male	1992-06-17	69000.00	2025-11-05 22:51:31
9	Meera	meera@example.com	Female	1989-09-05	77000.00	2025-11-05 22:51:31

7) ORDER BY

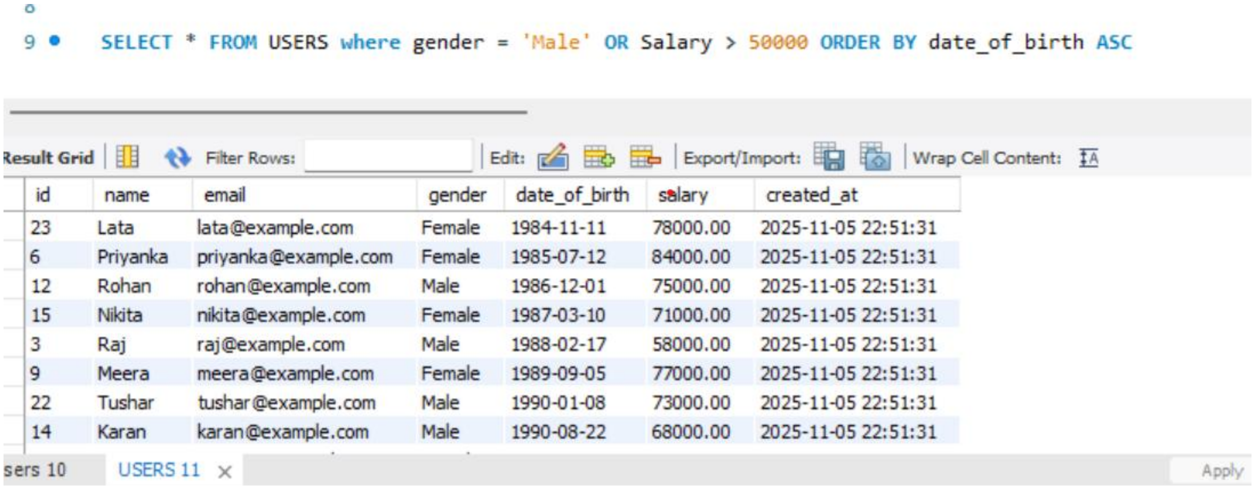
The ORDER BY clause is used to sort the result set in ascending (ASC) or descending (DESC) order based on one or more columns.

Query: SELECT * FROM users WHERE gender = ‘female’ OR salary > 50000 **ORDER BY** date_of_birth **DESC/ASC**;



```
SELECT * FROM USERS where gender = 'female' OR Salary < 90000 ORDER BY date_of_birth DESC
```

id	name	email	gender	date_of_birth	salary	created_at
10	Ishaan	ishaan@example.com	Male	2001-10-02	45000.00	2025-11-05 22:51:31
4	Sneha	sneha@example.com	Female	2000-08-09	50000.00	2025-11-05 22:51:31
19	Ritika	ritika@example.com	Female	1999-05-05	52000.00	2025-11-05 22:51:31
13	Zoya	zoya@example.com	Female	1998-01-15	54000.00	2025-11-05 22:51:31
24	Yash	yash@example.com	Male	1997-06-06	64000.00	2025-11-05 22:51:31
7	Aisha	aisha@example.com	Female	1997-03-25	56000.00	2025-11-05 22:51:31
16	Manav	manav@example.com	Male	1996-11-29	61000.00	2025-11-05 22:51:31
20	Imran	imran@example.com	Male	1995-07-30	63000.00	2025-11-05 22:51:31



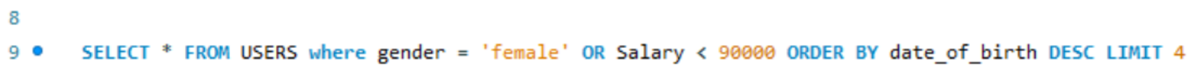
```
SELECT * FROM USERS where gender = 'Male' OR Salary > 50000 ORDER BY date_of_birth ASC
```

id	name	email	gender	date_of_birth	salary	created_at
23	Lata	lata@example.com	Female	1984-11-11	78000.00	2025-11-05 22:51:31
6	Priyanka	priyanka@example.com	Female	1985-07-12	84000.00	2025-11-05 22:51:31
12	Rohan	rohan@example.com	Male	1986-12-01	75000.00	2025-11-05 22:51:31
15	Nikita	nikita@example.com	Female	1987-03-10	71000.00	2025-11-05 22:51:31
3	Raj	raj@example.com	Male	1988-02-17	58000.00	2025-11-05 22:51:31
9	Meera	meera@example.com	Female	1989-09-05	77000.00	2025-11-05 22:51:31
22	Tushar	tushar@example.com	Male	1990-01-08	73000.00	2025-11-05 22:51:31
14	Karan	karan@example.com	Male	1990-08-22	68000.00	2025-11-05 22:51:31

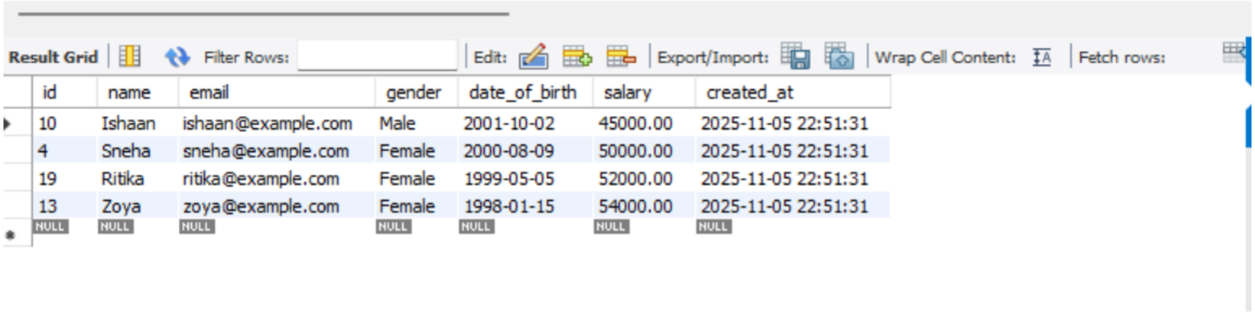
8) LIMIT

The LIMIT clause restricts the number of rows returned by a query.
It’s often used to view only a small subset of results.

Query: SELECT * FROM users WHERE gender = ‘female’ OR salary > 50000 **ORDER BY** date_of_birth **DESC/ASC LIMIT** 2;



```
SELECT * FROM USERS where gender = 'female' OR Salary < 90000 ORDER BY date_of_birth DESC LIMIT 4
```

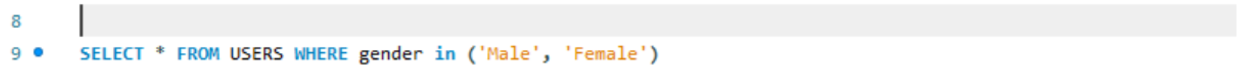


id	name	email	gender	date_of_birth	salary	created_at
10	Ishaan	ishaan@example.com	Male	2001-10-02	45000.00	2025-11-05 22:51:31
4	Sneha	sneha@example.com	Female	2000-08-09	50000.00	2025-11-05 22:51:31
19	Ritika	ritika@example.com	Female	1999-05-05	52000.00	2025-11-05 22:51:31
13	Zoya	zoya@example.com	Female	1998-01-15	54000.00	2025-11-05 22:51:31
NULL	NULL	NULL	NULL	NULL	NULL	NULL

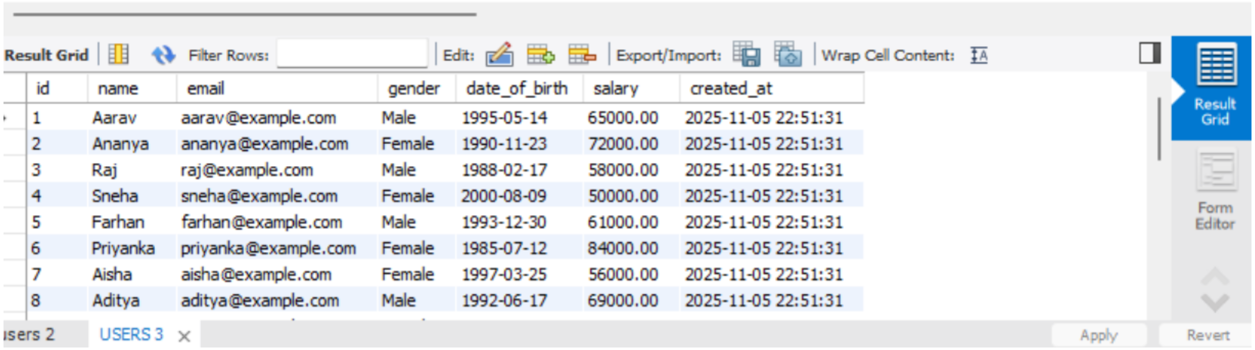
9) IN

The IN operator is used to match a column against multiple possible values in a single query.
It works like multiple OR conditions.

Query: SELECT * FROM users WHERE gender IN (‘Male’, ‘Female’);



```
SELECT * FROM USERS WHERE gender in ('Male', 'Female')
```

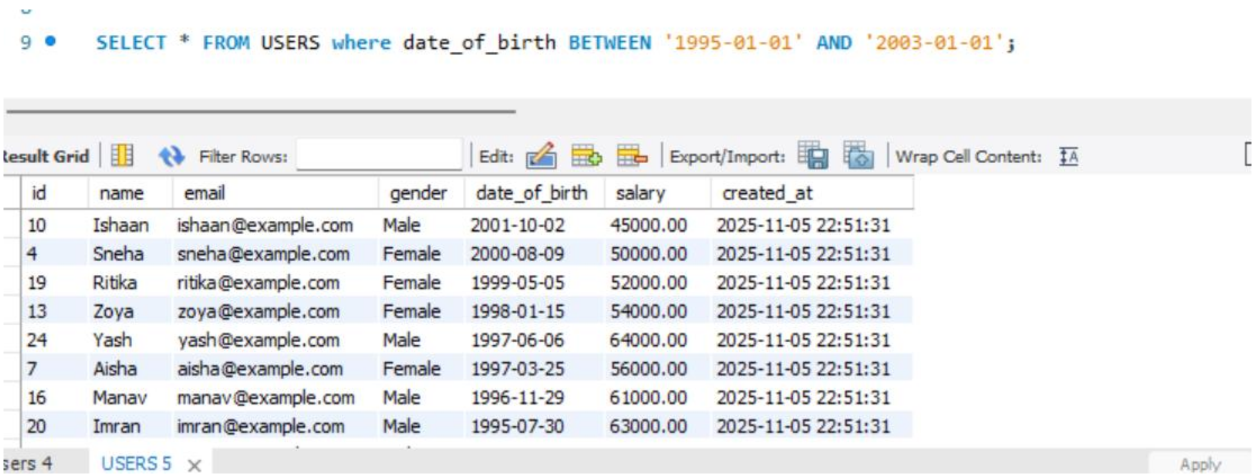


id	name	email	gender	date_of_birth	salary	created_at
1	Aarav	aarav@example.com	Male	1995-05-14	65000.00	2025-11-05 22:51:31
2	Ananya	ananya@example.com	Female	1990-11-23	72000.00	2025-11-05 22:51:31
3	Raj	raj@example.com	Male	1988-02-17	58000.00	2025-11-05 22:51:31
4	Sneha	sneha@example.com	Female	2000-08-09	50000.00	2025-11-05 22:51:31
5	Farhan	farhan@example.com	Male	1993-12-30	61000.00	2025-11-05 22:51:31
6	Priyanka	priyanka@example.com	Female	1985-07-12	84000.00	2025-11-05 22:51:31
7	Aisha	aisha@example.com	Female	1997-03-25	56000.00	2025-11-05 22:51:31
8	Aditya	aditya@example.com	Male	1992-06-17	69000.00	2025-11-05 22:51:31

10) BETWEEN

The BETWEEN operator is used to filter results within a range of values (inclusive).
It can be used with numbers, text, or dates.

Query: SELECT * FROM users WHERE date_of_birth BETWEEN ‘1995-01-01’ AND ‘2003-01-01’;



```
SELECT * FROM USERS where date_of_birth BETWEEN '1995-01-01' AND '2003-01-01';
```

id	name	email	gender	date_of_birth	salary	created_at
10	Ishaan	ishaan@example.com	Male	2001-10-02	45000.00	2025-11-05 22:51:31
4	Sneha	sneha@example.com	Female	2000-08-09	50000.00	2025-11-05 22:51:31
19	Ritika	ritika@example.com	Female	1999-05-05	52000.00	2025-11-05 22:51:31
13	Zoya	zoya@example.com	Female	1998-01-15	54000.00	2025-11-05 22:51:31
24	Yash	yash@example.com	Male	1997-06-06	64000.00	2025-11-05 22:51:31
7	Aisha	aisha@example.com	Female	1997-03-25	56000.00	2025-11-05 22:51:31
16	Manav	manav@example.com	Male	1996-11-29	61000.00	2025-11-05 22:51:31
20	Imran	imran@example.com	Male	1995-07-30	63000.00	2025-11-05 22:51:31

11) SQL Query execution order

Here is the typical order of execution in SQL:

- FROM and/or JOIN clause. (Identify the tables to retrieve data from. Joins are processed here, Combine data from multiple tables.)
- WHERE clause. (Filter rows based on conditions.)
- GROUP BY clause. (Group rows that have the same values in specified columns.)
- HAVING clause. (Filter groups created by GROUP BY.)
- SELECT statement. (Choose which columns or expressions to display.)
- DISTINCT clause. (Remove duplicate rows (if specified).)
- ORDER BY clause. (Sort the results.)
- LIMIT and/or OFFSET clause. (Return only a subset of rows.)

Query:

```
SELECT department, COUNT(*) AS total_employees
FROM employees
WHERE salary > 50000
GROUP BY department
HAVING COUNT(*) > 5
ORDER BY total_employees DESC
LIMIT 10;
```